ArchiVZ

A product of the VZ Technical Library.
Issue # 2 - 5th May 1994.
By Leslie Milburn.

Editorial

Welcome to the Second Issue. Let me first apologize for not including the Annual Indexes for ETI and EA with the last issue, the reason being that they would not fit in a standard envelope. Those people who do want them just write and let me know and I will send these separately.
I will be including some of the smaller Magazine Indexes with future newsletters. I will also be taking some of the more interesting articles from various sources and modifying them to be VZ specific.

My thanks goes to those who sent in programs, ideas and provided some of the missing items for the **T.L.** The response to the First issue was very pleasing, I was suprised to receive a few letters in the first week. There have been a couple of good suggestions but more on these later.

Anyway, enough of this until next time!

The VZ Technical Library

The main question I have been asked is regarding Copyright issues. Firstly let me say that this newsletter is not subject to copyright so feel free to copy it and pass it around.

The rules for distribution of copies of items held in the **T.L.** depend whether it is software of literature.

(1) **Software:-**

(i) If the software does not display a Copyright message or its source code does not contain any indications of its origins, it deemed to be **FREE**.

(ii) If the author can be identified, written permission is sought by myself to allow their software to be distributed.
If permission is granted the software is **FREE**.
If the author at first refuses to give permission then I will attempt to do a deal. Hopefully I will then be able to distribute it **FREE**. Otherwise the software is not Public Domain and therefore will not be freely distributed.

(iii) If the author is not in Australia or New Zealand, regardless of whether a Copyright message is displayed or not, if the software is more than 3 years old, I will deem it to be **FREE**.

(2) **User Group Newsletters/Magazines:-**

(i) Newsletters of User Groups which no longer exist are deemed **FREE**.

(ii) Magazines which are no longer available are deemed **FREE**.

(iii) I will provide copies of requested topics from Newsletters and Magazines still in production.

I hope that the above finally clears up any confusion. Please note that **FREE** generally does not include Postage costs

Below is an updated list of items currently held in the **T.L.** You will notice that it now includes more Magazines and Newsletters.

Magazines *(Photocopies of articles only provided)* :-

Magazine	Abbreviation	First Issue	Last Issue
All Magazine articles regarding the VZ200/300 Computers	-	-	-
Australian Personal Computer Magazine	APC	Vol 1 # 8	Vol 14 #10
Your Computer Magazine	YC	Vol 1 #1	Dec '92
Electronics Today International	ETI	Vol 6 #6	April '90
Electronics Australia	EA	Vol 38 #3	Vol 55 #4
Elektor Electronics	EE	190	219
ComputerCraft/MicroComputer Journal	MC	April '91	Feb '94
Australian Electronics Monthly	AEM	July '85	Nov '88
P.C. Games	PCG	Vol 1 #4	Vol 3 #1
Computes Gazette for VIC	VIC	Vol 1 #1	Vol 4 #1
80 Micro/80 Microcomputing	80	4/82, 5/82, 10/82, 9/83, 11/83, 12/83, 1/85 only	
Micro 80	M80	Vol 1 #1 - Vol 4 #1, Vol 4 #7 only.	
H & E Computronics	HE	#45, #46, #48, #52, #56, #57 only	
80 US/Basic Computing	BC	Vol V #9, Vol VI #7 only	
Dr Dobbs Journal	DOB	#134, #137, #150, #164, #200 only	
Alternative Computing Magazine	ALT	#1, #2 only	
Chips 'n' Bits	CNB	#6, #7, #8 only	
UK Electronics Today International	UK	Jan - Mar '93, Nov '93 - Mar '94 only	
Byte Magazine	BYT	Nov '80, Dec '81, Mar '82, Aug '82, Jan '83, Dec '83, Jun '84, Jan '85, Nov '85, May '86, Jun '86, Sept '89, Nov '89, Sept '90, Nov '90, Dec '90 only	
MegaComp	ME	#3 only	
Computer Fun & Games	CFG	#2 only	
Silicon Chip	SC	Vol 1 #2, Vol 1 #7, Vol 1 #10, Vol 1 #11, Vol 1 #12, Vol 1 #14, Vol 2 #5, Vol 3 #8, Vol 4 #8, Vol 4 #9, Vol 4 #11, Vol 5 #10, Vol 5 #11, Vol 6 #5, Vol 6 #8, Vol 6 #12, Vol 7 #1 only	
PC User Specials			
PC User Tips & Tricks		'93	
Your Computer Specials			
Learning with YC		#1, #3, #4, #5, #6	
Program Present		'87 - FREE with YC	
Bumper Book of Programs from YC		#1, #2, #3	
Tech Tips Digest from YC			
PC Dictionary from YC		'93	
EA Specials			
Getting into Microprocessors		1977	
The Sound Revolution - Audio CD		1985	
A Basic Guide to Colour TV and VCRs		1992	
ETI Specials			
Computers and Computing		#1, #2, #3, #4	
Test Gear		#1, #2, #3	
ETI Circuits		#1, #2, #4, #5	
Saturday Arvo Projects			
15 Saturday Arvo Projects			
25 Top Projects			
Electronics its easy		#1, #2	
Electronic projects for your Car			
ETI Magazine		Vol 4 #8	
APC Specials			
Business Computing Survival Guide Pt1		'86	
Viatel Directory		#4 '86	
Benchtest Special		'87	
Buyers Guide to Software		'87, '88, '90	
Storage Special			
Macintosh Supplement		May '88	
Printer Supplement		'90	
10th Birthday Supplement			
Communications Special			
Programming Supplement			

User Group Newsletters *(Photocopies only provided)* :-

NewsLetter	Issue	Notes
DSE VZ200 Interface	#2 - #3	Only 3 issues were produced. #1 Missing.
DSE Comput	#1 - #5	Complete Set. Also New Zealand #1 which combined Aus #1 & #2.
LE'VZ200/300 OOP	#1 - #27	Complete Set
VEE ZEE News	#1 - #12	Complete Set.
Micro Magic	#1 - #6	Complete Set.
Studio Ad Lib	#1 - #6	Complete Set.
VZCOMPU200/300	#1 - #4	Complete Set.
Hunter Valley Journal	#1 - #44	Complete Set. Still going strong.
VZ Down Under	#1 - #40	Complete Set.
DiskMag	#1 - #4	Complete Set.
Out West VZ200 User Group	#1 - #2	Complete Set.
WAVZ	#1	Complete Set.
Visual Display Unit	#4 only	Luigi Chiodo was Editor. At least 7 issues produced.
VZ Link (NZ)	#1 - #48 only	Peter Hill was Editor. 73 issues produced. Missing #44, #49 - #73
J&R Software	None.	Last known tape letter was #3 in Jan '85.
XILOG (NZ)	None.	At least 8 issues produced (#8 dated Aug '85).
LYSCO	None.	At least 10 issues produced. Went from Nov '84 to July '86
Christchurch VZ User Group		Went from July '84 to April '88

Software:-

For a limited time only, **Gary McCleary** has kindly given his permission for me to freely distribute the following of his programs:-

Code Maker/Breaker
Maths Pack
Number Challenge
Word Challenge
Graphic Golf
Ten Pin Bowling

plus a couple of others which I have temporarily forgotten (I do not have his letter in front of me). If you want any of his software then you had better get in quick.

Other Software that is freely available is as follows:-

Description	Author
All VZ200/300 Programs	Dick Smith.
The Quickwrite Word Processor/Text Editor	L. Milburn.
Diskops - All Versions	L. Milburn.
Mouse Driver	L. Milburn.
Keyboard Driver	L. Milburn.
Extended BASIC (XB)	R. Harrison

To Be Continued.....

Due to time restrictions I have had to hold back the promised list of Books until next issue. I will also be including a **Full** list of Software that I have, indicating which are Public Domain (**FREE**).

Below is a list of those magazines which are either missing from the **T.L.** or a damaged copy is held. If you have any of these which you don't want let me know.

Name	Issue/Volume	Notes
	Needed	
PCG	19 July 1984 Vol 1 #1	Back Issues were available from:-
PCG	Aug 1984 Vol 1 #2	77 Glenhuntly Road
PCG	Oct 1984 Vol 1 #3	Elwood
PCG	Dec 1984 Vol 1 #5	VIC 3164
PCG	Feb 1985	(03) 531 8411
AEM	Nov 1986	
AEM	Dec 1987	
AEM	Aug 1988	
AEM	Dec 1988 onwards.	Was December 1988 the last issue??
EA	Pre June 1976	Not desperate for these.
EA	Jan, March 1977	
EA	Aug, Sept 1979	
EA	Nov 1989	
EA	May, Aug 1990	
EA	93 (except April) & onwards	Not desperate for these.
ETI	Pre June 1976 (except November 1974)	ETI commenced sometime in 1971. Not desperate for these.
ETI	Feb, Mar, May, Jun 1977	
ETI	July, Aug, Sept 1979	
ETI	April, Nov 1988	Contains VZ Articles
ETI	Mar, April, May, Oct 1989	Contains VZ Articles
ETI	Feb, Mar, May 1990	May 1990 was the last issue of ETI before it merged with Electronics Australia.
APC	Vol 1 issue 1 - Vol 1 issue 7 (inclusive)	
APC	Dec 1989	
APC	Jan, Dec 1990	
APC	Nov 1993 and onwards.	
YC	Sept 1983	
YC	Jun, Aug - Oct 1984	
YC	Jan, July, Nov, Dec 1985	
YC	Mar - July, Sept - Nov 1986	
YC	Jan - Mar, May - Nov 1987	
YC	Jan - Feb, Apr - May, Aug, Sept, Nov, Dec 1988	
YC	Jan, Apr 1989	
YC	May - Dec 1990	
YC	Jan - May, Aug - Nov 1991	
YC	Feb - Nov 1992	
YC	All 1993 and onwards.	
M80	Vol 4 #2 - Vol 4 #6.	
M80	Vol 4 #8 onwards.	
CFG	#1, #3 onwards	How many were produced??
	Damaged	
EA	Nov 1984	Cover Damaged
EA	Dec 1988	Very Damaged
AEM	July, Aug, Sept, Oct, Nov 1985	Water Damaged
AEM	Jan, Feb, Apr, May, June 1986	Water Damaged
YC	Dec 1987	Cover Damaged

Trading Post

This section is dedicated to the VeeZedders who wish to sell their setup. I shall only list their name, contact number, hardware list and what price they want for it. In some cases some of this information may be missing.

Name	Contact Number	Price	Hardware List
Les Brennan		Make an offer.	VZ300 & Power supply, Floppy Disk Drive, Disk Controller, Printer Interface, 64K Ram, 16K Ram, VZ Terminal Cartridge, VZ RTTY Cartridge, Word Pro Cartridge, 2 DR10 Datasettes
L. Brill		$50.00 plus postage	VZ200 16K Ram, Datasette, Joysticks, Software Tapes
Barry Clarke		-	Datasette $20 RTTY/Printer Int. $5 Plotter Rolls (6) $15 Software Tapes - Call
Lew Glickman		$95.00	VZ300, 16K Ram, Datasette, Software Tapes, Books
Brian Hanlon		Best Offer	VZ300, 16K Ram, Datasette, Software Tapes
Leslie Milburn		Phone for list.	Duplicate copies of magazines and books held in the Technical Library. Also original Dick Smith Software Tapes.

Please also refer to **issue #1** for other items for trading. I will print an updated list every **three** issues.

Buyers: If you buy any items from the above people could you please let me know so that I can update the list.

Sellers: If you do sell any of your equipment can you pass their name and address on to me so that I can keep in contact with as many VZ Users as possible.

BASIC Programming

Before everybody gets up and yells Oh no! not another tutorial on BASIC, let me just establish from the outset that this section is **NOT** for that purpose.

Generally, there are two types of VZ users who receive this publication, those who have owned their VZ for many years and those who have bought a VZ as a cheap way to learn computing. So as you can see there is a wide gap in knowledge. I hope (major hint coming up) that those who have done a lot of BASIC programming will be generous enough to submit a program.
To overcome doing the same thing that other Newsletters have done in the past, I have decided that all programs presented in this section will adhere to the following guidelines:-

(1) I assume that everyone can at least type in a BASIC program and has some knowledge of the fundamentals of the language.
If not then I recommend that you read **Introduction to Computing** which I consider to be excellent and is how I learnt. The **T.L.** has some original copies for sale at $5.00 plus pp or a photocopy could be provided for postage alone.

(2) The programs will fit an unexpanded VZ300.

(3) My presented "solution" will be able to be compiled by the **LASCOM Basic Compiler** (the what?) - more on this later.

The aim of this section is mainly educational. All programs I will present here will be discussed in detail. We will examine the coding style, design of the program and the use of algorithms. Hopefully everyone will appreciate the necessity for clear coding, which is fully commented and also be able to decide upon the most efficient algorithms to use in any particular case. Remember, the program must be efficient but also easily maintainable by yourself or other people.
Please note that quite often there is more than one solution for any programming task. I would welcome any comments or alternative solutions from anyone, however please make sure that they meet points 2 & 3 above.

It is important for those people who do submit programs **NOT** to be embarrassed if theirs is criticised. I hope they will realise the benefits of such discussion.

Now after all that long drawn out blurb we come to the program under the magnifying glass for this issue. The honour of submitting the first program goes to **Ray** of **Warwick**, **QLD**. This program is a **Word Puzzle Generator** which was originally written for the **Microbee**. Now what we must first note is that as different machines have slightly different versions of BASIC it was not just a task of typing the program straight in, some translation was needed. Following is the resultant translated program.
Comments in **Bold** are what I have added to make your understanding easier and I have separated certain sections. I have also included a list of variables used in the program. These were not in the original submission.

[As a side note I would be interested to hear **YOUR** views debating **Translation of Programs** versus **Program Rewrite**]

----------Start of Program Listing----------

```
10    REM WORD PUZZLE GENERATOR
20    CLS:PRINT@32,"THIS IS A WORD PUZZLE
      GENERATOR,";
30    PRINT"PLEASE HAVE YOUR PRINTER
      READY  ON LINE."
40    CLEAR5000:B$=""
```

/* Get User Input. */

```
50    PRINT:PRINT"WHAT ARE THE 'X' AND
      'Y' DIMENS-";
60    PRINT"IONS (LENGTHS OF SIDES -
      MAXIMUMSIZE 35 X 30)"
70    INPUT"HOW WIDE";X'INPUT"HOW
      LONG";Y
```

```
80    IFX>35ORY>30THEN70
90    IFX>YTHENU0=XELSEU0=Y
100   INPUT"NUMBER OF WORDS";N
110   DIMA$(X,Y),N$(N+2),X$(N),L(N,3)
```

/* Main Program Loop. */

```
120   GOSUB1400:Q8=1:PRINT:PRINT"COMPUTER
      SORTING...PLEASE WAIT!!"
130   GOSUB1300:PRINT:FORI1=1TON:GOTO150
140   L(I1,1)=A:L(I1,2)=B:L(I1,3)=D:NEXTI1:
      GOTO900
150   C0=0:S$=X$(I1):Z=LEN(S$)
160   GOSUB800:C0=C0+1:IF(C0/100)<>INT(C0/100)
      THEN180
170   PRINTQ8" TRY #"C0
180   IFC0<1000THEN220
190   Q8=Q8+1:SOUND21,3:IFQ8=100THENSTOP
200   GOTO130
```

/* Determine Direction of Current Word. */

```
220   'ON D GOTO230,290,360,420,490,550,610,670
221   IFD=1THEN230
222   IFD=2THEN290
223   IFD=3THEN360
224   IFD=4THEN420
225   IFD=5THEN490
226   IFD=6THEN550
227   IFD=7THEN610
228   IFD=8THEN670
```

/* Down. */

```
230   IFB+Z>YTHEN160
240   FORI=1TOLEN(S$):A1=A:B1=B+I-1:
      GOSUB1100:GOSUB1200
250   IFL$=B$ORL$=N1$THEN270
260   GOTO160
270   NEXTI
280   FORI=1TOLEN(S$):GOSUB1100:
      A$(A,B+I-1)=N1$:NEXTI:GOTO140
```

/* Down & Left. */

```
290   IFB+Z>YORA-Z<1THEN160
300   FORI=1TOLEN(S$):A1=A-(I-1):
      B1=B+I-1:GOSUB1100:GOSUB1200
310   IFL$=B$ORL$=N1$THEN330
320   GOTO160
330   NEXTI
340   FORI=1TOLEN(S$):GOSUB1100:
      A$(A-(I-1),B+I-1)=N1$:NEXTI
350   GOTO140
```

/* Left. */

```
360   IFA-Z<1THEN160
370   FORI=1TOLEN(S$):A1=A-(I-1): B1=B:
      GOSUB1100:GOSUB1200
380   IFL$=B$ORL$=N1$THEN400
390   GOTO160
400   NEXTI
410   FORI=1TOLEN(S$):GOSUB1100:
      A$(A-(I-1),B)=N1$:NEXTI:GOTO140
```

/* Left & Up. */

```
420   IFA-Z<1ORB-Z<1THEN160
430   FORI=1TOLEN(S$):A1=A-(I-1):B1=B-(I-1):
      GOSUB1100:GOSUB1200
440   IFL$=B$ORL$=N1$THEN460
450   GOTO160
460   NEXTI
470   FORI=1TOLEN(S$): GOSUB1100:
      A$(A-(I-1),B-(I-1))=N1$:NEXTI
480   GOTO140
```

/* Up. */

```
490   IFB-Z<1THEN160
500   FORI=1TOLEN(S$):A1=A:B1=B-(I-1):
      GOSUB1100:GOSUB1200
510   IFL$=B$ORL$=N1$THEN530
520   GOTO160
530   NEXTI
540   FORI=1TOLEN(S$):GOSUB1100:
      A$(A,B-(I-1))=N1$:NEXTI:GOTO140
```

/* Right & Up. */

```
550   IF(A+Z)>XOR(B-Z)<1THEN160
560   FORI=1TOLEN(S$):A1=A+I-1:B1=B-(I-1):
      GOSUB1100:GOSUB1200
570   IFL$=B$ORL$=N1$THEN590
580   GOTO160
590   NEXTI
600   FORI=1TOLEN(S$):GOSUB1100:
      A$(A+I-1,B-(I-1))=N1$:NEXTI:GOTO140
```

/* Right. */

```
610   IFA+Z>XTHEN160
620   FORI=1TOLEN(S$):A1=A+I-1:B1=B:
      GOSUB1100:GOSUB1200
630   IFL$=B$ORL$=N1$THEN650
640   GOTO160
650   NEXTI
660   FORI=1TOLEN(S$):GOSUB1100:
      A$(A+I-1,B)=N1$:NEXTI:GOTO140
```

/* Right & Down. */

```
670   IFA+Z>XORB+Z>YTHEN160
680   FORI=1TOLEN(S$):A1=A+I-1:B1=B+I-1:
      GOSUB1100:GOSUB1200
690   IFL$=B$ORL$=N1$THEN710
700   GOTO160
710   NEXTI
720   FORI=1TOLEN(S$):GOSUB1100:
      A$(A+I-1,B+I-1)=N1$:NEXTI:GOTO140
```

/* Get Coords and Direction for Word. */

```
800   A=RND(X)
810   B=RND(Y)
820   D=RND(8)
830   RETURN
```

/* Print Puzzle. */

```
900   LPRINTCHR$(17)
910   FORJ=1TOY
920   FORI=1TOX
930   IFA$(I,J)=""THENA$(I,J)=CHR$(RND(26)+64)
940   LPRINTA$(I,J)" ";
950   NEXTI:LPRINT:NEXTJ:LPRINT:GOSUB1000:
      LPRINT:CLS:GOTO30
1000  PRINT"PLEASE WAIT WHILE I SORT THE"
1005  PRINT"WORDS INTO ALPHABETICAL ORDER,"
1010  PRINT"IT COULD TAKE A WHILE."
1015  PRINT"FOR 40-50 WORDS PLEASE ALLOW    5
      MINS OR SO."
1020  GOSUB2000:LPRINT"THE HIDDEN WORDS ARE:"
1030  FORK=1TONSTEP3:LPRINTN$(K);
      TAB(20)N$(K+ 1); TAB(40)N$(K+2)
1060  NEXTK:RETURN
1100  N1$=MID$(S$,I,1):RETURN

1200  L$=A$(A1,B1):RETURN

1300  FORI=1TOX:FORJ=1TOY:A$(I,J)="":NEXTJ,I:
      RETURN

1400  IF(INP(0)AND1)=0THEN1410
1405  PRINT@480,"YOUR PRINTER IS NOT ON
      LINE.";:GOTO1400
1410  LPRINT"I HOPE THIS PUZZLE REALLY STUMPS
      YOU.":LPRINT
1415  GOSUB1700:LPRINTCHR$(19)
1420  FORI=1TON
1430  PRINT"WORD #";I:INPUTX$(I):N$(I)=X$(I):
      GOSUB1500:GOSUB1600
1435  IFRW=1THEN1430
1440  IFLEN(X$(I))>U0THEN1430
1450  NEXTI:RETURN
```

/* Strip Non-Alphabetic Characters. */

```
1500  CH$=X$(I)
1510  FORC=1TOLEN(CH$)
1520  CH=ASC(MID$(CH$,C,1))
1530  IFCH<65ORCH>90THEN1560
1540  NEXT:X$(I)=CH$
1550  RETURN
1560  CH$=LEFT$(CH$,C-1) +
      RIGHT$(CH$,LEN(CH$)-C)
1570  GOTO1510
```

/* Check for the Existance of Word. */

```
1600  FORQ=1TOI-1
1605  IFI<2THENRW=0:RETURN
1610  IFX$(I)=X$(Q)THENRW=1:RETURN
1620  NEXT
1630  RW=0:RETURN
```

/* Get and Print Puzzle Heading. */

```
1700  PRINT"DO YOU WANT TO ADD A HEADING OR"
1710  PRINT"COMMENT? IF SO, PUT IT NOW."
1720  PRINT"IF NOT, JUST PRESS <<RETURN>>."
1730  INPUTZ$: IFZ$=""THENRETURN
1740  LPRINTZ$:RETURN
```

/* Sort the List of Words. */

```
2000  LPRINTCHR$(19):FORJ=1TON-1
2010  FORI=1TON-J
2020  IFN$(I)>N$(I+1)THENT$=N$(I):N$(I)=N$(I+1):
      N$(I+1)=T$
2030  NEXT:NEXT:LPRINTCHR$(17):RETURN
```

----------End Of Program Listing----------

Variables used in Program Above.

Variable	Description
B$	Empty or blank string.
X	Width of puzzle - User Input.
Y	Height of puzzle - User Input.
U0	Largest of the puzzle dimensions.
N	Number of words in puzzle - User Input.
A$()	Puzzle word map array.
N$()	Array containing words entered by User.
X$()	Array containing copy of words after non-alphabetic characters stripped.

L()	Array containing Coords and direction of each word in Puzzle.
Q8	Retry counter. No of every 1000 retries made to place current word in puzzle.
I1	Current word number.
C0	Retry counter. Counts each word placement attempt until 1000.
S$	Current word to be placed in puzzle.
Z	Length of current word.
D	Randomly selected direction of current word.
A, B	Randomly selected X and Y coords for start of current word.
A1, B1	X and Y coords for each character of the current word.
N1$	Current character in current word.
L$	Character at coords A1, B1 in puzzle.
I, J, K	General purpose loop variables.
Q	Loop variable used when checking if word already entered.
RW	Indicates whether current word is already entered.
CH$	Current character when testing for non-alphabetic characters in word.
C	Loop variable used when stripping non-alphabetic characters.
CH	Ascii value of CH$ (above)
Z$	Storage for Heading/Comment
T$	Temporary variable used when sorting.

My Comments:

Well I must say that when the program is presented in the format above it is quite readable. Just wait until you have typed it in!
Anyway, my first gripe is that the original code had no comments. Even the few I have provided help immensely when trying to understand it.
The structure of the program is quite good, but there are a couple of places where **GOTO**'s are used when better programming could have avoided them, try and use **GOTO**'s sparingly. Always try and limit the range of **GOTO**'s to the immediate subroutine.
There are places where the same sequence of code is being repeated with just slight variations. This could have easily been made into a subroutine.
Many variables have been used and sometimes their names are not obvious as to what function they are performing. Try and give variables a decent name, but remember that our version of BASIC only counts the first two characters for uniqueness.
My Last point is that you should always indicate somewhere in the program who wrote it or where the listing originally came from. Currently I am having great trouble identifying a lot of programs which I have obtained. This simple task would have made my job a lot easier.

My Solution (?):

Below is my "solution" You will see that I have commented every subroutine providing information about input variables and return values. You may even take this one step futher and indicate what variables (if any) are changed by the routine.

When creating this program I changed the functionality slightly. The main changes I made were as follows:-

(1) The Words are sorted as they are entered. This means that user input is slower but you don't have to wait nearly ten minutes while the words are sorted.
(2) A Facility to print the answer has been provided. This was done by not populating the puzzle map with the random characters but choosing them whilst printing. This means that you will not be able to print the identical puzzle twice - although this is not really an issue.
(3) The Maximum puzzle dimensions and number of words are fixed.
(4) You will also notice from the code that the puzzle map is not held in an array but stored in unused area of screen memory. The main reason for this is so that the program can be compiled with the LASCOM Compiler.

```
10    'WORD PUZZLE VERSION 2
20    'BY L. MILBURN
30    '10TH MARCH 1994
40    '
50    '*************************
60    'INITIALISATION
65    '
70    CLEAR1000:DIM WRD$(49)
80    PUZZLE=29184
85    MODE(1):CLS ' REM QUICKEST WAY TO
      CLEAR THE PUZZLE AREA.
90    CURRLAST=0
95    FALSE=0:TRUE=-1
100   '*************************
110   'MAIN PROGRAM LOOP
115   '
120   GOSUB 200 'GET DIMENSIONS
130   GOSUB 400 'GET WORDS
140   SOLN=0:GOSUB 1950 'PRINT PUZZLE
150   GOSUB 2300 'PRINT WORDS
160   PRINT:INPUT"DO YOU WANT TO PRINT
      THE ANSWER";CURRWRD$
170   CH$=LEFT$(CURRWRD$,1)
175   IF CH$ <> "Y" THEN GOTO 190
180   SOLN=1:GOSUB 1950 'PRINT PUZZLE
      SOLUTION
190   END
200   '*************************
210   'GET PUZZLE DIMENSIONS
215   '
220   'UPON EXIT:
230   ' X = PUZZLE WIDTH
240   ' Y = PUZZLE HEIGHT
250   ' NOWRDS = NO OF WORDS
260   '
270   CLS:PRINT"PLEASE ENTER THE
      DIMENSIONS OF"
280   PRINT"THE WORD PUZZLE."
290   INPUT"ENTER WIDTH (1-30)";X
300   IF X<1 OR X>30 THEN GOTO 290
310   INPUT"ENTER HEIGHT (1-35)";Y
320   IF Y<1 OR Y>35 THEN GOTO 310
330   INPUT"ENTER NO. OF WORDS (1-
      50)";NOWRDS
340   IF NOWRDS<1 OR NOWRDS>50 THEN 330
350   RETURN
400   '*************************
410   'GET WORDS
420   '
430   'UPON ENTRY:
440   ' X = WIDTH OF PUZZLE
450   ' Y = HEIGHT OF PUZZLE
460   ' NOWRDS = NO WORDS TO GET
470   '
475   'UPON EXIT:
480   ' PUZZLE = PTR TO A MAP OF WORD
      POSITIONS
490   ' WRD$ = LIST OF ENTERED WORDS
500   FOR WRD=0 TO NOWRDS-1
510   PRINT"ENTER WORD #";WRD+1
520   INPUT CURRWRD$
525   IF CURRWRD$="" THEN GOTO 510
530   GOSUB 600 'VALIDATE WORD
540   IF RC <>TRUE THEN GOTO 510
550   GOSUB 900 'PLACE WORD IN PUZZLE
560   NEXT WRD
570   RETURN
600   '*************************
610   'VALIDATE WORD
620   '
630   'UPON ENTRY:
640   ' CURRWRD$ = WORD TO CHECK
650   ' CURRLAST = CURR NO WORDS
660   '
670   'UPON EXIT:
680   ' RC = INDICATES IF WORD OK
690   '
700   RC=TRUE
710   IF CURRLAST=0 THEN GOTO 770
720   FOR I=0 TO CURRLAST
730   IF WRD$(I)=CURRWRD$ THEN
      RC=FALSE:RETURN
740   IF WRD$(I)>CURRWRD$ THEN 800
750   NEXT
760   'ADD TO LIST
770   WRD$(CURRLAST)=CURRWRD$
780   CURRLAST=CURRLAST+1
790   RETURN
800   'INSERT INTO LIST
810   FOR J=CURRLAST TO I+1 STEP -1
820   WRD$(J)=WRD$(J-1)
830   NEXT
840   WRD$(I)=CURRWRD$
850   CURRLAST=CURRLAST+1
860   RETURN
900   '*************************
910   'PLACE WORD IN PUZZLE
920   '
930   'UPON ENTRY:
940   ' CURRWRD$ = CURRENT WORD
950   ' PUZZLE = PTR TO MAP OF WORD POSITIONS
960   ' X = PUZZLE WIDTH
970   ' Y = PUZZLE HEIGHT
1000  ATTEMPT=0
1010  J=LEN(CURRWRD$)
1020  A=RND(X)-1:B=RND(Y)-1:D=RND(8)
1030  GOSUB 1710 'CHECK LIMITS
1040  IF RC=FALSE THEN GOTO 1080
1050  ATTEMPT=ATTEMPT+1
```

```
1060  IF ATTEMPT<>100 THEN GOTO 1020 'TRY
      AGAIN
1070  PRINT"UNABLE TO PLACE";
1080  PRINTCURRWRD$;
1075  PRINT" IN PUZZLE":END
1080  FOR I=1 TO J
1090  GOSUB 1200 ' GET CHAR FROM MAP
1100  IF CH$="" THEN GOTO 1140
1110  WCH$=MID$(CURRWRD$,I,1)
1120  IF CH$=WCH$ THEN GOTO 1140
1130  GOTO 1020 'TRY AGAIN
1140  NEXT 'CHECK NEXT CHAR
1150  FOR I=1 TO J
1160  CH$=MID$(CURRWRD$,I,1)
1170  GOSUB 1400 'PUT CHAR INTO MAP
1180  NEXTI
1190  RETURN
1200  '*************************
1210  'GETCHAR
1220  '
1230  'UPON ENTRY:
1240  ' A = X DIRECTION COORD
1250  ' B = Y DIRECTION COORD
1260  ' D = WORD DIRECTION
1270  '
1280  'UPON EXIT:
1290  ' CH$ = CHAR AT COORDS A,B
1300  '
1310  GOSUB 1520 'GET COORDS
1320  CH$=CHR$(PEEK(PUZZLE+CRD))
1325  IFCH$=CHR$(0) THEN CH$=""
1330  RETURN
1400  '*************************
1410  'PUTCHAR
1420  '
1430  'UPON ENTRY
1440  ' A = X DIRECTION COORD
1450  ' B = Y DIRECTION COORD
1460  ' D = WORD DIRECTION
1470  ' CH$ = CHARACTER TO STORE
1480  '
1490  GOSUB 1520 'GET COORDS
1500  POKE PUZZLE+CRD,ASC(CH$)
1510  RETURN
1520  '*************************
1530  'GET COORDS
1540  '
1550  'UPON ENTRY:
1560  ' A = X DIRECTION COORD
1570  ' B = Y DIRECTION COORD
1580  ' D = WORD DIRECTION
1590  '
1600  'UPON EXIT:
1610  ' CRD = ADJUSTED COORDS
1620  '
1630  IF D=1 CRD=A+30*(B+I-1):RETURN
1640  IF D=2 CRD=(A-I-1)+30*(B+I-1):RETURN
1650  IF D=3 CRD=(A-I-1)+30*B:RETURN
1660  IF D=4 CRD=(A-I-1)+30*(B-I-1):RETURN
1670  IF D=5 CRD=A+30*(B-I-1):RETURN
1680  IF D=6 CRD=(A+I-1)+30*(B-I-1):RETURN
1690  IF D=7 CRD=(A+I-1)+30*B:RETURN
1700  CRD=(A+I-1)+30*(B+I-1):RETURN
1710  '*************************
1720  'CHECKBOUNDS
1730  '
1740  'UPON ENTRY:
1750  ' X = PUZZLE WIDTH
1760  ' Y = PUZZLE HEIGHT
1770  ' A = X DIRECTION COORD
1780  ' B = Y DIRECTION COORD
1790  ' D = WORD DIRECTION
1795  ' J = CURR WORD LENGTH
1800  '
1810  'UPON EXIT:
1820  ' RC = INDICATES IF COORDS ARE WITHIN
      BOUNDS OF PUZZLE
1830  '
1840  IF D=1 RC=(B+J)>Y:RETURN
1850  IF D=2 RC=(B+J)>Y OR (A-J)<1:RETURN
1860  IF D=3 RC=(A-J)<1:RETURN
1870  IF D=4 RC=(A-J)<1 OR (B-J)<1:RETURN
1880  IF D=5 RC=(B-J)<1:RETURN
1890  IF D=6 RC=(A+J)>X OR (B-J)<1:RETURN
1900  IF D=7 RC=(A+J)>X:RETURN
1910  RC = (A+J)>X OR (B+J)>Y:RETURN
1950  '*************************
1960  'PRINT PUZZLE
1970  '
1980  'UPON ENTRY:
1990  ' X = PUZZLE WIDTH
2000  ' Y = PUZZLE WIDTH
2010  '
2020  GOSUB 2100 ' CHECK PRINTER
2025  GOSUB 2180 ' GET USER HEADING
2030  FOR I=0 TO Y-1
2040  FOR J=0 TO X-1
2045  CH$=CHR$(PEEK(PUZZLE+(J+30*I)))
2050  IF SOLN=0 THEN 2065
2055  IF CH$=CHR$(0) THEN CH$="."
2060  GOTO 2070
2065  IF CH$=CHR$(0) THEN CHR$(RND(26)+64)
2070  LPRINTCH$;" ";
2080  NEXTJ
2090  LPRINT:NEXTI:RETURN
2100  '*************************
2110  'CHECK PRINTER
2120  '
2130  IF(INP(0)AND1)=0 THEN 2160
2140  PRINT@480,"YOUR PRINTER IS NOT ONLINE";
2150  GOTO2130
2160  PRINT@480,"                    ";
2170  RETURN
```

```
2180  '*************************
2190  'GET HEADING
2200  '
2210  PRINT:PRINT"IF YOU WISH TO HAVE A
      HEADING"
2220  PRINT"ENTER ONE NOW. IF NOT JUST"
2230  INPUT"PRESS RETURN";CH$
2240  GOSUB 2100 'CHECK PRINTER
2250  LPRINTCH$:LPRINT:RETURN
2300  '*************************
2310  'PRINT WORDS
2320  '
2330  'UPON ENTRY:
2340  ' NOWRDS = NO OF WORDS
2350  '
2360  J=0
2370  LPRINT:LPRINT"THE HIDDEN WORDS ARE:-"
2380  FOR I=0 TO NOWRDS-1
2390  LPRINTWRD$(I);TAB(20*(J+1));
2400  J=J+1:IFJ=3 LPRINT:J=0
2410  NEXTI
2420  LPRINT
2430  RETURN
```

----------End Of Program Listing----------

Variables used in Program Above.

Variable	Description
PUZZLE	Pointer to the word storage area.
FALSE	What it says.
TRUE	What it Says.
SOLN	Indicates whether to print the puzzle or the solution.
X	Width of puzzle - User Input.
Y	Height of Puzzle - User Input.
NOWRDS	Number of words in puzzle - User Input.
WRD$	Array containing words - sorted on input.
RC	"Return Code" - General variable for results of subroutine calls.
WRD	Current word number.
CURRWRD$	Current word.
I, J	General purpose loop variables (routine specific).
ATTEMPT	The number of attempts to place word in puzzle.
A,B	Randomly selected X and Y coords for start of current word.
D	Randomly selected direction of current word.
CRD	Coords for current character of current word in word storage area.
CH$, WCH$	General purpose character storage (routine specific).

Machine Code Programming

This section is only very small for this issue. Basically this is just to let you know what I plan to do in future issues regarding Machine/Assembly Code programming.

Starting next issue there will be two new sections. One will be for advanced Assembly Language Programmers, the other will be for people using BASIC who would just like to understand a little bit about Machine Code programming. I had hoped to start the latter this issue but I have simply run out of time and pages.

For the Advanced Programmers I have decided to do something not done in previous VZ Newsletters, a Major Software project. Basically we are going to write a new **Operating System** from scratch. This will also include the writing of tools to help us in our quest. I anticpate that this will take many issues and be very challenging. We will be targeting a VZ with a 16K Memory Expansion as our base, but will be providing support for up to 255 banks of switched memory. This means we shall be able to support up to 4 Megabytes of memory. Our project will also include writing device drivers for new hardware addons - such as 3.5 inch 720K Disk Drives, a Real Time clock, Expanding the capacity of the Current Disk system. I hope this has whet your appetite and has suddenly triggered a whole host of ideas from you. If so let me know.

A Review of the LASCOM BASIC Compiler

As I mentioned earlier in the BASIC Programming section, all presented solutions will be able to be compiled with this Compiler.

First of all what is a Compiler? Well, Basically speaking (excuse the pun) a Compiler performs translation of a program into Machine code, yes that's it! Easy eh!

The LASCOM Compiler was written in Germany by Frank Nicklisch. Laserlink bought the Australian Distribution Rights and imported it. The cost from memory was about $40.00 which I believe was very reasonable. By the way, as far as I know, these rights have since lapsed.

When loaded the Program asks you to specify some configuration parameters, mainly concerning the size or arrays and strings. After a time I found this very annoying because once specified there was no way to change them unless you switched off and on again and then reloaded the compiler. Anyway, this is just a minor problem.

Considering the size of the compiler, the number of command it supports is very rich. There are of course some limitations such as no disk drive support. However, some additional facilities are provided including allowing you to embed hex bytes into the program. An adventurous programmer will then be able to "talk" to the disk drive easily.

All BASIC commands are supported except the following:-

VERIFY, CSAVE, CONT, LIST, LLIST, NEW, PRINT# and INPUT#.

Other noticeable restrictions are:-

(1) Floating point (or real number) variables are not available
(2) LOG, SIN, COS, TAN, ATN, EXP are not allowed due to (1)
(3) Only fixed single dimension arrays are allowed
(4) You cannot have code which has more than one NEXT per FOR although this may be programatically correct!

Due to the compiler being in memory, you can only input or load a BASIC program which is up to 9K in size.
To get around this a facility has been provided which will allow you to link programs together.

Yes, a lot of features are not provided, what do you want - Blood!
If you code your program correctly, once compiled it will run anywhere between 20 to 200 times faster. This is especially useful if you are using graphics. The Word Puzzle solution is a classic example. When run under normal BASIC the printer is slowed down because it is constantly waiting for the VZ to send it information (the printer has an 8K buffer). With the compiled version of the program the puzzle was sent immediately and control is returned to the User. Also noticeable is that the sort on input feature is instant and as such goes unnoticed.

Submissions

I will accept submissions in virtually ANY format, either on Disk or Tape. Failing that I will also accept a printed copy of short items which MUST be clear. Please remember that in this case I will have to re-enter your article/program.
I also have a TRS80 and have access to an IBM. So basically if you send it I can use it!!

Other VZ Publications/User Groups

Hunter Valley Journal Editor: Joe Leon
Address: 35 Tighes Terrace,
Tighes Hill
NSW 2297

This is an excellent publication containing a wealth of information about the VZ. It contains many Hardware and Software projects for the beginner and expert alike.
I recommend that you subscribe and support this journal. Subscription rates are approximately $18-00 for six issues. I would also advise that you order all Back Issues for inclusion in your personal VZ library.

Other VZ Groups

News from the News Zealand Front. If you remember last issue I mentioned that I had heard that two VZ User Groups could still be operating in New Zealand. I have since had confirmation that one of these groups is now defunct. I am still awaiting news about the other.

Groups of Interest

Starting with this issue I shall include information about any User Groups and Bulletin Board Systems (BBS) which could be of interest to VZ Users. Please note that I do not have time to investigate the accuracy of this information and would appreciate hearing from anybody who has tried to make contact.

Adelaide Micro User Group (caters for TRS80) Contact: Max Morris
Phone: (08) 386 2284

Micro 80 Pty Ltd (used to cater for TRS80 & VZ) Phone: (08) 211 7244

End Notes

Here is a little piece of information for anybody wishing to have a go at upgrading their VZ to use a faster chip.
The speed of a standard VZ200/300 is just below 4 MHz. Apparently a 25 MHz version of the Z80 is now being made in the U.S. While I have not had time to investigate whether there is a supplier in Australia, one User has informed me that a 20MHz Z80 is available from G.E.C.. For those interested, the phone number is (02) 638 1888. The part number is Z84C0020PEC and costs $30.00 a chip when stocks of TEN are ordered. Remember that once you have upgraded, tape and disk operation will be affected!!
For those of you who do not understand what this upgrade will achieve, what it means is that the effective speed of the VZ would be almost equal to that of a new IBM PC and faster than probably 90% of the older IBM's.

I would welcome any feedback on the **VZ Quick Reference Card** I have included with this issue.

If you wish to get in touch with me, my address is

2/124 Railway Parade, Mortdale NSW 2223 Australia.

or phone (02) 580 0492 between 6 and 8 pm most days.
Please leave a message if I do not answer. If there is one thing that bugs me it is people who do not leave messages!!